ELSEVIER

Theoretical Computer Science 166 (1996) 147–172

Theoretical Computer Science

Meta-resolution: An algorithmic formalisation

Jean-Denis Fouks[a,*], Jean-Claude Spehner[b,1]

[a] *IRCOM-SIC URA 356, Université de Poitiers, 86022 Poitiers Cedex, France*
[b] *Université de Mulhouse, Faculté de Sciences et Technologie, F-68093 Mulhouse, France*

Received January 1993; revised July 1995
Communicated by M. Nivat

Abstract

We study the Tseitin formulae (Tseitin, 1970) on which resolution is known to be inefficient. We first give a new proof of the satisfiability condition and precise the number of solutions if any and otherwise the minimal contradictory subformula. Then, defining a new data structure, we introduce the meta-argument of that proof in the Davis–Putnam and Loveland procedure (Loveland, 1978) and obtain a new algorithm for SAT especially efficient on these formulae.

0. Introduction

The question of the complexity of resolution, one of the most efficient methods to solve the SAT problem, has been studied by many researchers. From this point of view, an important family of contradictions was introduced by Cook and Reckhow [4]. These formulae encoding the pigeonhole principle were proved to be intractable for general resolution by Haken [9].

Another important method to obtain non-trivial contradictions was defined by Tseitin [13]. Heuristically, its idea was the following: Among the 2^n clauses on $\underline{A} = \{\underline{1}, \ldots, \underline{n}\}$, 2^{n-1} contain an odd number of negative literals (the odd block) and 2^{n-1} an even number (the even block). Any valuation satisfying the 2^{n-1} clauses of the odd block (resp. of the even block) takes the value "true" an even (resp. odd) number of times. For instance, if $\underline{A} = \{\underline{1}, \underline{2}\}$, the two clauses $\{1', 2\}$ and $\{1, 2'\}$ are both satisfied by the valuations ($\underline{1} \rightarrow$ "true", $2 \rightarrow$ "true") and ($\underline{1} \rightarrow$ "false", $2 \rightarrow$ "false"); the two

* This paper was originally presented at the Journées sur le Problème de la Satisfiabilité, Barbizon, France, 1991. The Editor-in-Chief would like to acknowledge the help of Olivier Dubois (Université Pierre et Marie Curie, Paris) in the publication of this paper.

* Corresponding author. E-mail: fouks@zeus.univ-poitiers.fr.
[1] E-mail: spehner@univ-mulhouse.fr.

SSDI 0304-3975(95)00155-7

clauses $\{1,2\}$ and $\{1',2'\}$ are both satisfied by the valuations ($\underline{1} \to$ "true", $\underline{2} \to$ "false") and ($\underline{1} \to$ "false", $\underline{2} \to$ "true").

Now, let $G = (X, U)$ be a simple connected graph whose set U of edges is indexed by a set A of atoms. Each vertex x of G is associated by its boundary $f(x)$ to a subset of $\underline{A}$ and consequently with two "blocks" of $2^{|f(x)|-1}$ clauses. To obtain a formula, we choose for each vertex x one of the two blocks and give to x the opposite parity. For instance, let G be given by Fig. 1. For x, we choose the odd block $\{1,2,3'\}$, $\{1,2',3\}$, $\{1',2,3\}$ and $\{1',2',3'\}$ and so x is even. For y, we choose the odd block $\{1,4'\}$ and $\{1',4\}$ and so y is even. For z, we choose the even block $\{3,5\}$ and $\{3',5'\}$ and so z is odd. For t, we choose the even block $\{2,4,5\}$, $\{2,4',5'\}$, $\{2',4,5'\}$ and $\{2',4',5\}$ and so t is odd.

Let v be a valuation satisfying every clause of such a formula. For each vertex x, the parity of the number of edges $u_{\underline{i}} \in f(x)$ such that $v(\underline{i}) =$ "true" is equal to the parity of x. As each edge belongs to two boundaries, we can see that the obtained formula is satisfiable if and only if the number of odd vertices is even. For instance, the formula obtained above is satisfied by the valuation:

($\underline{1} \to$ "false", $\underline{2} \to$ "false", $\underline{3} \to$ "false", $\underline{4} \to$ "false", $\underline{5} \to$ "true").

If we replace the even block $\{\{3,5\},\{3',5'\}\}$ by the odd one $\{\{3,5'\},\{3',5\}\}$, the new formula is contradictory. So this method associates to a graph G a formula $C(G)$ encoding the principle "If S is a sum where each term appears twice then S must be even"; linking clauses with subgraphs, Tseitin has obtained a counting method for the number of distinct clauses in regular trees and proved the exponentiality of regular resolution.

Using Haken's method and a specific family of graphs derived from those of Margulis [11], Urquhart [14] has proved the exponentiality of general resolution for the corresponding Tseitin formulae. Another proof of probabilistic nature has been given by Chvatal and Széremedy [3].

In the present paper, after a generalisation of Tseitin's method to graphs with multiple edges and loops, we give a new proof of the satisfiability condition, determine the number of solutions if any and precise the minimal contradictory subformula

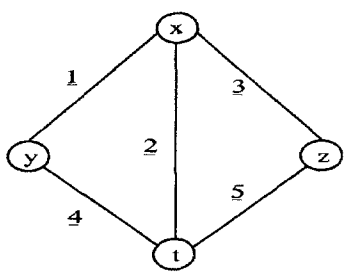


Fig. 1.

otherwise. That proof leads us to a new concept of meta-resolution encoding in a single step a complete resolution tree of two blocks. Using an appropriate data structure, we define the meta-resolution trees which provide short refutations of several kinds of contradictions including Tseitin's contradictory formulae and their minimal contradictory subformulae. Finally, integrating our meta-resolution trees in the Davis–Putnam and Loveland procedure [10], we obtain a new algorithm for the SAT problem solving Tseitin's formulae in polynomial time.

0.1. Definitions and notations

We consider a set $\underline{A}$ of "atoms"; each atom $\underline{i}$ of $\underline{A}$ supports two opposite literals i and i'.

A finite set of literals whose supporting atoms are distinct from one another is called a *clause*. For each clause c, we denote by c' the symmetric clause $\{\ell, \ell' \in c\}$, by $\underline{c}$ the set of supporting atoms and by $\mathrm{sgn}(c)$ the parity of the number of negative literals. For instance, if $c = \{1, 2', 3]$ then $c' = \{1', 2, 3'\}$, $\underline{c} = \{\underline{1}, \underline{2}, \underline{3}\}$ and $\mathrm{sgn}(c) = -$. The empty clause is denoted by Λ. A finite set of clauses is called a *formula*. Let $\underline{B}$ be a subset of $\underline{A}$. The formula $\{c, \underline{c} = \underline{B}, \mathrm{sgn}(c) = \varepsilon\}$ denoted by $\mathscr{B}^{\varepsilon}(\underline{B})$ is called the *block with support* $\underline{B}$ *and sign* ε. For instance, $\mathscr{B}^{+}(\{\underline{1}, \underline{2}, \underline{3}\}) = \{\{1, 2, 3\}, \{1, 2', 3'\}, \{1', 2, 3'\}, \{1', 2', 3\}\}$. By convention, $\mathscr{B}^{+}(\emptyset) = \{\Lambda\}$ and $\mathscr{B}^{-}(\emptyset) = \emptyset$. If a block is denoted by $\mathscr{B}$ then its support (resp. sign) will be referred as $\underline{\mathscr{B}}$ (resp. $\mathrm{sgn}(\mathscr{B})$).

A map from $\underline{A}$ to $\{+, -\}$ is called a *valuation* and is identified with a set of literals. Let c be a clause and v be a valuation. If $c \cap v \neq \emptyset$ then we say that v *satisfies* c. Let F be a formula and v be a valuation; if, $\forall c \in F$, v satisfies c and v is called a *solution* of F. A formula F is said to be *contradictory* (resp. *satisfiable*) if its set of solutions is empty (resp. not empty). For instance, the formula $\{\{1\}, \{1'\}\}$ is contradictory and the formula $\{\{1', 2\}, \{1, 2'\}\}$ is satisfiable.

1. Formula defined by a graph

1.1. Alternated graphs

The graphs we shall consider have a finite set X of "vertices" and a family $U = \{u_i\}_{i \in I}$ of "edges" which are unordered pairs $\{x, y\}$ of vertices. $\forall x \in X$, the set of edges (resp. loops) containing x is called the *boundary* (resp. *interior boundary*) of x denoted by $f_G(x)$ (resp. $\mathrm{fi}_G(x)$); $f_G(x) \backslash \mathrm{fi}_G(x)$ is called the *exterior boundary* of x denoted by $\mathrm{fe}_G(x)$. When G is obvious, we write $f(x)$, $\mathrm{fi}(x)$ and $\mathrm{fe}(x)$. The set of connected components of G is denoted by $\mathrm{co}(G)$.

Definition 1.1. (i) $G = (X, U, \sigma)$ is called an *alternated graph* on A if:
(X, U) is a graph where A is the set of indices of U.
σ is a map from X to $\{+, -\}$, called the sign.

(ii) Let $H = (Y, V)$ be a subgraph of G. The product $\prod_{y \in Y} \sigma(y)$ is called the *sign* of H and is denoted by $\sigma(H)$. H is said to be *even* if $\sigma(H) = +$ and *odd* otherwise.

In graphic representations, vertices of sign $+$ (resp, $-$) will be marked $\circ$ (resp. $\bullet$).

Definition 1.2 (*Deletion of edges*). Let ℓ be a literal on $\underline{A}$; the map $\sigma \backslash \ell$ from X to $\{+, -\}$ is defined as follows:

If ℓ is positive or if $\underline{\ell}$ is a loop then $\sigma \backslash \ell = \sigma$.

Otherwise, $\sigma \backslash \ell(x) = -\sigma(x)$ if $x \in \underline{\ell}$ and $\sigma \backslash \ell(x) = \sigma(x)$ if $x \notin \underline{\ell}$.

(a) The alternated graph $(X, U \backslash \{\underline{\ell}\}, \sigma \backslash \ell)$ is said to be obtained by *deletion* of ℓ in G and is denoted by $G \backslash \ell$.

(b) As for any ℓ_1, ℓ_2, $(\sigma \backslash \ell_1) \backslash \ell_2 = (\sigma \backslash \ell_2) \backslash \ell_1$, we can naturally define $\sigma \backslash c$ for each clause c on $\underline{A}$. The alternated graph $(X, U \backslash \underline{c}, \sigma \backslash c)$ is said to be obtained by *deletion* of c in G and is denoted by $G \backslash c$. If $\underline{B}$ is a subset of $\underline{A}$, the graph $(X, U \backslash \underline{B})$ will be denoted by $G \backslash \underline{B}$.

An example is given in Fig. 2.

As $4 = \{y, z\}$, $\sigma \backslash 4'(y) = -\sigma(y)$ and $\sigma \backslash 4'(z) = -\sigma(z)$. As $\underline{1} = \{x, x\}$ is a loop, $\sigma \backslash 1'(x) = \sigma(x)$.

Definition 3 (*Cyclomatic notions*). An edge u is called a *cycle edge* if $G \backslash u$ has the same number of connected components than G and an *isthmus* otherwise. The number $\nu(G) = |U| - |X| + |\mathrm{co}(G)|$ is called the *cyclomatic number* of G. For every graph G, $\nu(G)$ is a non-negative integer (see [1]).

Definition 1.4. (i) Let $G = (X, U, \sigma)$ be an alternated graph and x a vertex of G. The set of clauses

$$|c = d_1 \cup d_2,\ d_1 \in \mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x)),\ d_2 = \mathrm{fi}(x)\}$$

is said to be *associated with* x and is denoted by $G(x)$.

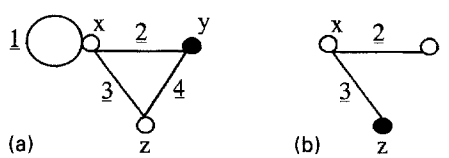


Fig. 2. (a) A grah G. (b) The graph $G \backslash \{1', 4'\}$.

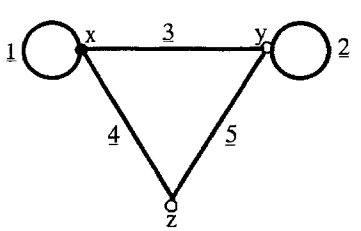


Fig. 3.

(ii) Let G be an alternated graph. The set $\bigcup_{x\in X} G(x)$ is caled the *formula defined* by G and is denoted by $C(G)$.

Example. Let G be given by Fig. 3. $G(x) = \{\{1,3,4\},\{1',3,4\},\{1,3',4'\},\{1',3',4'\}\}$, $G(y) = \{\{2,3,5'\},\{2',3,5'\},\{2,3',5\},\{2',3',5\}\}$ and $G(z) = \{\{4,5'\},\{4',5\}\}$. $C(G) = G(x)\cup G(y)\cup G(z)$.

Remark. In Tseitin's original work [13], only simple graphs were used (i.e. without multiple edges and loops).

2. Satisfiability condition

Lemma 2.1 (Restriction Lemma). *Let $G = (X, U, \sigma)$ be an alternated graph and let $\underline{i}$ be an edge of G.*

(a) *Let $v \in \mathrm{sol}(C(G))$; its restriction to $\underline{A}\setminus\{\underline{i}\}$ belongs to $\mathrm{sol}(C(G\setminus i'))$ if $v(\underline{i}) = +$ and to $\mathrm{sol}(C(G\setminus i))$ if $v(\underline{i}) = -$.*

(b) *If $\underline{i}$ is a loop then every solution of $C(G\setminus i')$ is the restriction of two solutions of $C(G)$; otherwise, every solution of $C(G\setminus i')$ (resp. $C(G\setminus i)$) is the restriction of exactly one solution v of $C(G)$ and $v(\underline{i}) = +$ (resp. $-$).*

Proof. (a) Let x be a vertex of G and let $c \in (G\setminus i')(x)$; the case where $c \in (G\setminus i)(x)$ is similar. If $x \notin \underline{i}$ then $c \in C(G)$ and the restriction of v satisfies c.

If $x \in \underline{i}$ then, by definition of $(G\setminus i')(x)$, $\underline{c} = f_{G\setminus\underline{i}}(x)$ and $\mathrm{sgn}(c\,|\,\mathrm{fe}_{(G\setminus\underline{i})}(x)) = -(\sigma\setminus i')(x)$: if $\underline{i}$ is a loop then $\underline{c}\cup\{\underline{i}\} = f_G(x)$ and $\mathrm{sgn}((c\cup\{i\})\,|\,\mathrm{fe}_G(x)) = \mathrm{sgn}((c\cup\{i'\})\,|\,\mathrm{fe}_G(x)) = \mathrm{sgn}(c\,|\,\mathrm{fe}_{(G\setminus\underline{i})}(x)) = -\sigma(x)$. So, $C(G) \supset \{c\cup\{i\}, c\cup\{i'\}\}$ and the restriction of v satisfies c. If $\underline{i} \in \mathrm{fe}(x)$ then $\underline{c}\cup\{\underline{i}\} = f_G(x)$ and $\mathrm{sgn}((c\cup\{i'\})\,|\,\mathrm{fe}_G(x)) = -\mathrm{sgn}(c\,|\,\mathrm{fe}_{(G\setminus\underline{i})}(x)) = (\sigma\setminus i')(x) = -\sigma(x)$. So, $c\cup\{i'\}$ belonging to $C(G)$ is satisfied by v and if $v(\underline{i}) = +$ its restriction satisfies c.

(b) Let w be a solution of $C(G\setminus i')$; w is the restriction of two valuations v_1 and v_2. Let x be a vertex of G and let $c \in G(x)$. If $\underline{i} \notin \underline{c}$ then w and consequently v_1 and v_2 satisfy c.

If $\underline{i} \in \underline{c}$ then $\underline{c} = f_G(x)$ and $\mathrm{sgn}(c\,|\,\mathrm{fe}_G(x)) = -\sigma(x)$; we denote by d the restriction $c\,|\,f_{G\setminus\underline{i}}(x)$. If $\underline{i}$ is a loop then $\mathrm{sgn}(d\,|\,\mathrm{fe}_{G\setminus\underline{i}}(x)) = \mathrm{sgn}(c\,|\,\mathrm{fe}_G(x)) = -\sigma(x) = -(\sigma\setminus i')(x)$ and $d \in C(G\setminus i')$. So d and c are satisfied by w hence by v_1 and v_2.

If $i \in \mathrm{fe}_G(x)$ we examine the two possible cases:

(i) If $i' \in c$ then $d = c\setminus\{i'\}$ and $\mathrm{sgn}(d\,|\,\mathrm{fe}_{G\setminus\underline{i}}(x)) = -\mathrm{sgn}(c\,|\,\mathrm{fe}_G(x)) = \sigma(x) = -(\sigma\setminus i')(x)$. So d and c are satisfied by w hence by v_1 and v_2.

(ii) If $i \in c$ then let d_0 be the restriction of w' to $f_{G\setminus\underline{i}}(x)$; as d_0 is not satisfied by w it does not belong to $C(G\setminus i')$ and $\mathrm{sgn}(d_0\,|\,\mathrm{fe}_{G\setminus\underline{i}}(x)) = (\sigma\setminus i')(x) = -\sigma(x)$. As $\mathrm{sgn}(d_0\cup\{i\}\,|\,\mathrm{fe}_G(x)) = \mathrm{sgn}(d_0\,|\,\mathrm{fe}_{G\setminus\underline{i}}(x)) = -\sigma(x)$, $d_0\cup\{i\} \in G(x)$ and is satisfied by the valuation v_1 such that $v_1(\underline{i}) = +$ but is not satisfied by v_2.

So, if i is a loop the two extensions of w belong to $\mathrm{sol}(C(G))$ but otherwise, only the extension v such that $v(\underline{i}) = +$ is a solution of $C(G)$. The case of a solution of $C(G\backslash i)$ is similar.

Lemma 2.2 (Separation Lemma). *Let G be an alternated graph and let $\underline{i}$ be an edge of G. Then*

$$|\mathrm{sol}(C(G))| = |\mathrm{sol}(C(G\backslash i))| + |\mathrm{sol}(C(G\backslash i'))|.$$

Proof. $\mathrm{sol}(C(G))$ is the disjoint union of $\mathrm{sol}_1 = \{v \in \mathrm{sol}(C(G)),\ v(\underline{i}) = +\}$ and $\mathrm{sol}_2 = \{v \in \mathrm{sol}(C(G)),\ v(\underline{i}) = -\}$.

By Restriction Lemma, $|\mathrm{sol}_1| = |\mathrm{sol}(C(G\backslash i'))|$ and $|\mathrm{sol}_2| = |\mathrm{sol}(C(G\backslash i))|$.

Theorem 2.3 (Satisfiability Theorem). *If all the connected components of an alternated graph $G = (X, U, \sigma)$ are even then $|\mathrm{sol}(C(G))| = 2^{\nu(G)}$; otherwise $C(G)$ is contradictory.*

A similar result appears in [13] but without the determination of the number of solutions if any.

Proof. We proceed by induction on $|U|$.

Basis. Let $\underline{1}$ be the only edge of U. If $\underline{1} = \{x, x\}$ is a loop then

when $\sigma(x) = +$, $\quad C(G) = \emptyset$, $\quad \mathrm{sol}(C(G)) = \{\{1\}, \{1'\}\}$,

when $\sigma(x) = -$, $\quad C(G) = \{\{1\}, \{1'\}\}$ is contradictory.

If $\underline{1} = \{x, y\}$ is not a loop then

when $\sigma(x) = \sigma(y)$, $\quad C(G) = \{1\}$ or $\{1'\}$, $\quad |\mathrm{sol}(C(G))| = 1$,

when $\sigma(x) = -\sigma(y)$, $\quad C(G) = \{\{1\}, \{1'\}\}$ is contradictory.

Induction step. Let $|U| > 1$, $\underline{i}$ be an edge of U and let H be the connected component of G containing $\underline{i}$.

If H is odd so are $H\backslash i$ and $H\backslash i'$ and they both contain an odd component. We conclude by induction hypothesis that $C(H\backslash i)$ and $C(H\backslash i')$ are contradictory and by the separation lemma that $C(H)$ and $C(G)$ are contradictory.

If all the connected components of G are even, we examine two cases:

(i) If $\underline{i}$ is an isthmus, $H\backslash i$ has two even (resp. odd) components if and only if $H\backslash i'$ has two odd (resp. even) components. By the induction hypothesis, we conclude that only one of the two formulae $C(G\backslash i)$ and $C(G\backslash i')$ is satisfiable, the number of its solutions being $2^{\nu(G\backslash i)}$; by separation Lemma we obtain $\mathrm{sol}(C(G)) = 2^{\nu(G\backslash i)} = 2^{\nu(G)}$.

(ii) If $\underline{i}$ is a cycle edge, $H\backslash i$ and$H\backslash i'$ are connected and even. By the separation lemma, we conclude that $|\mathrm{sol}(C(G)| = |\mathrm{sol}(C(G\backslash i))| + |\mathrm{sol}(C(G\backslash i'))|$; by the induction hypothesis, we obtain $\mathrm{sol}(C(G)) = 2^{\nu(G\backslash i)+1} = 2^{\nu(G)}$.

3. Minimality condition

In this part, G is an odd connected alternated graph.

3.1. Territories

Definition 3.1. Let c be a clause; an odd connected component of $G\setminus c$ is called a *territory* of c. As G is odd and connected, the number of territories of c is odd.

Examples. Let x be vertex of G; if $c\setminus G(x)$ then $(\{x\},\emptyset)$ is a territory of c but not always the only one. Let G be given by Fig. 4. The clause $c=\{1,2',3,4'\}$ belongs to $G(x)$ but admits as territories the three connected components of $G\setminus c$.

3.2. Simple clauses

Definition 3.2. A clause with only one territory is said to be *simple*; the territory of a simple clause c is denoted by $\mathrm{ter}(c)$.

Examples. If x is not a breaking point of G then every clause of $G(x)$ is simple. If G is given by Fig. 4 then the clause $\{1,2,3,4,5\}$ is simple.

Property 3.3. *Any subclause d of a simple clause c is simple.*

Proof. Every connected component of $G\setminus d$ is a disjoint union of connected components of $G\setminus c$ and an odd component of $G\setminus d$ contains an odd number of odd components of $G\setminus c$. If c is simple there is only one odd component of $G\setminus c$ and hence only one odd component of $G\setminus d$; so d is simple.

Theorem 3.4 (Inclusion Theorem). *Let x be a vertex of G and $\tau_x(G)$ be the alternated graph obtained from G by reversing $\sigma(x)$.*

(i) *A valuation v belongs to $\mathrm{sol}(C(\tau_x(G)))$ if and only if v' is simple and* $\mathrm{ter}(v')=(\{x\},\emptyset)$.

(ii) *If c is a simple clause such that $x\in\mathrm{ter}(c)$ then c' is included in $2^{\nu(G\setminus c)}$ solutions of* $\tau_x(G)$.

(iii) *The subset of simple clauses is the only minimal contradictory subformula of* $C(G)$.

(iv) *For any $x\in X$, $G(x)$ contains exactly $2^{\delta(x)}$ simple clauses.*

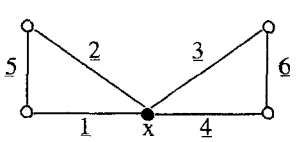


Fig. 4.

Proof. We denote by G^* the graph $\tau_x(G)$.

(i) If $v \in \mathrm{sol}(C(G^*))$ then by Restriction Lemma all the components of $G^* \setminus v'$ are even and hence $(\{x\}, \emptyset)$ is the only odd component of $G \setminus v'$. Conversely, if $(\{x\}, \emptyset)$ is the only odd component of $G \setminus v'$ then all the components of $G^* \setminus v'$ are even and by the Restriction Lemma $v \in \mathrm{sol}(C(G^*))$.

(ii) As the only odd connected component of $G \setminus c$ contains x, all the components of $G^* \setminus c$ are even and by the satisfiability theorem $|\mathrm{sol}(C(G^* \setminus c))| = 2^{\nu(G \setminus c)}$. Now by the restriction lemma, the restriction to $A \setminus c$ defines a one-to-one mapping from the subset of solutions of $C(G^*)$ containing c' to the set of solutions of $C(G^* \setminus c)$.

(iii),(iv) In order to obtain a contradictory formula, we add clauses of $G(x)$ to $C(G^*)$. If v is a solution of $C(G^*)$ and if d is a clause of $G(x)$, d is not satisfied by v if and only if d is a subclause of v' which is simple by point (i); so to be useful, d needs to be a subclause of a simple clause hence needs to be simple. Let Φ be the disjoint union of the subsets $\{v \in \mathrm{sol}(C(G^*)), v' \supset d\}$ when d describes the set of simple clauses of $G(x)$. As $C(G)$ is contradictory, $\Phi = \mathrm{sol}(C(G^*))$ and by point (ii) each one of the subsets $\{v \in \mathrm{sol}(C(G^*)), v' \supset d\}$ contains $2^{\nu(G) - \delta(x)}$ elements. By the satisfiability theorem, we know that $|\Phi| = 2^{\nu(G)}$ and we conclude that the number of simple clauses of $G(x)$ is $2^{\delta(x)}$, each of them being necessary to obtain a contradiction.

4. Meta-resolution

4.1. Resolution

If $c_1 \cap c_2' = \{\ell\}$ then the clause $(c_1 \setminus \ell) \cup (c_2 \setminus \ell)$ denoted by $c_1 \triangledown c_2$ is called the *resolvent* of c_1 and c_2 and is said to be obtained by *annihilation* of $\underline{\ell}$. For instance, the resolvent of $\{1', 2, 3\}$ and $\{3', 4, 5\}$ is $\{1', 2, 4, 5'\}$ but $\{1', 3, 4'\}$ and $\{3', 4, 5'\}$ have no resolvent.

Resolution is the proof system adding recursively to a formula the resolvents defined by two of its clauses. By Robinson's theorem [12], a formula F is contradictory if and only if there exists a resolution of F producing the empty clause. The exponentiality of this method has been proved for pigeonhole formulae [9] and Tseitin's formulae [7, 14]. These formulae are not intrinsically hard but few efficient algorithms are known to solve them.

For pigeonhole formulae, a proof using cutting planes may be obtained in linear time; on Tseitin's formulae, cutting planes proofs seem to have exponential lengths, Lesniewski's algorithm mentioned in [14] is in $O(n^4)$ and extended resolution [13] is undeterministically in $O(n^3)$. The concept of meta-resolution, appearing implicitly in our proof of the satisfiability condition, leads to a more efficient method.

4.2. Meta-resolvent

Let $\mathscr{B}_1$ and $\mathscr{B}_2$ be blocks. The block with the symmetric difference $\mathscr{B}_1 \triangle \mathscr{B}_2$ for support and $-(\mathrm{sgn}(\mathscr{B}_1) \cdot \mathrm{sgn}(\mathscr{B}_2))$ for sign is called their meta-resolvent denoted by $\mathscr{B}_1 \bigtriangledown \mathscr{B}_2$.

Theorem 4.1 (Comparison Theorem). *Let $\mathscr{B}_1$ and $\mathscr{B}_2$ be blocks. Every solution of $\mathscr{B}_1 \cup \mathscr{B}_2$ is a solution of $\mathscr{B}_1 \bigtriangledown \mathscr{B}_2$ and every solution of $\mathscr{B}_1 \bigtriangledown \mathscr{B}_2$ is either a solution of $\mathscr{B}_1 \cup \mathscr{B}_2$ or a solution of $\mathscr{B}^{-\mathrm{sgn}(\mathscr{B}_1)}(\underline{\mathscr{B}}_1) \cup \mathscr{B}^{-\mathrm{sgn}(\mathscr{B}_2)}(\underline{\mathscr{B}}_2)$. Consequently, any clause consequence of $\mathscr{B}_1 \cup \mathscr{B}_2$ contains either an initial clause or a clause of the meta-resolvent.*

Proof. We denote by ε_1 and ε_2 the parity of $\mathscr{B}_1$ and $\mathscr{B}_2$, respectively. Let v be a valuation and let d_1, d_2 and d_0 denote the restriction of v' to $\underline{\mathscr{B}}_1 \setminus \underline{\mathscr{B}}_2$, $\underline{\mathscr{B}}_2 \setminus \underline{\mathscr{B}}_1$ and $\underline{\mathscr{B}}_1 \cap \underline{\mathscr{B}}_2$, respectively. If v is a solution of $\mathscr{B}_1 \cup \mathscr{B}_2$ then $d_1 \cup d_0$ and $d_2 \cup d_0$ which are not satisfied by v do not belong to $\mathscr{B}_1 \cup \mathscr{B}_2$. As their supports are, respectively, $\underline{\mathscr{B}}_1$ and $\underline{\mathscr{B}}_2$, $\mathrm{sgn}(d_1 \cup d_0) = -\varepsilon_1$ and $\mathrm{sgn}(d_2 \cup d_0) = -\varepsilon_2$. So, $\mathrm{sgn}(d_1 \cup d_2) = \mathrm{sgn}(d_1) \cdot \mathrm{sgn}(d_2) = \mathrm{sgn}(d_1) \cdot \mathrm{sgn}(d_0) \cdot \mathrm{sgn}(d_0) \cdot \mathrm{sgn}(d_2) = \mathrm{sgn}(d_1 \cup d_0) \cdot \mathrm{sgn}(d_2 \cup d_0) = \varepsilon_1 \cdot \varepsilon_2$ and $(d_1 \cup d_2) \notin \mathscr{B}_1 \bigtriangledown \mathscr{B}_2$. As it is the only clause with support $\underline{\mathscr{B}}_1 \triangle \underline{\mathscr{B}}_2$ which is not satisfied by v, we can conclude that v is a solution of $\mathscr{B}_1 \bigtriangledown \mathscr{B}_2$.

Conversely, let v be a solution of $\mathscr{B}_1 \bigtriangledown \mathscr{B}_2$. If v does not satisfy $\mathscr{B}^{-\varepsilon_1}(\underline{\mathscr{B}}_1)$ then $\mathrm{sgn}(d_1 \cup d_2) = \varepsilon_1 \cdot \varepsilon_2$, $\mathrm{sgn}(d_1 \cup d_0) = -\varepsilon_1$ and $\mathrm{sgn}(d_2 \cup d_0) = \mathrm{sgn}(d_1 \cup d_2) \cdot \mathrm{sgn}(d_1 \cup d_0) = -\varepsilon_2$. In that case, the only clause with support $\underline{\mathscr{B}}_1$ (resp. $\underline{\mathscr{B}}_2$) which is not satisfied by v belongs to $\mathscr{B}^{-\varepsilon_1}(\underline{\mathscr{B}}_1)$ (resp. $\mathscr{B}^{-\varepsilon_2}(\underline{\mathscr{B}}_2)$) and v is a solution of $\mathscr{B}_1 \cup \mathscr{B}_2$. Similarly, v is a solution of $\mathscr{B}^{-\varepsilon_1}(\underline{\mathscr{B}}_1) \cup \mathscr{B}^{-\varepsilon_2}(\underline{\mathscr{B}}_2)$ if it does no satisfy $\mathscr{B}_1$.

Now, if a clause d contains neither an initial clause nor a clause of the meta-resolvent then there exists a valuation v such that v' contains d but no initial clause. So, v is a solution of the initial set which does not satisfy d and d is not a consequence of $\mathscr{B}_1 \cup \mathscr{B}_2$.

Principle 4.2 (Comparison Principle). Let F_1 and F_2 be formulae such that $\mathrm{sol}(F_1) \subset \mathrm{sol}(F_2)$. In order to prove that F_1 is contradictory, it is sufficient to prove that F_2 is contradictory. So, when $\mathrm{sol}(F_1) \subset \mathrm{sol}(F_2)$, we write $F_1 < F_2$.

5. Block arithmetic

5.1. Products

(i) Let c be a clause and $\mathscr{B}$ be a block. If $\underline{c} \cap \underline{\mathscr{B}} = \emptyset$ then the set of clauses $\{c \cup d, d \in \mathscr{B}\}$ is called the *product* of c by $\mathscr{B}$ and is denoted by $c \cdot \mathscr{B}$. For instance, $\{3,4'\} \cdot \mathscr{B}^+(\{\underline{1},\underline{2}\}) = \{\{1,2,3,4'\},\{1',2',3,4'\}\}$.

Remark. For any clause c, its product by the empty block $\mathscr{B}^-(\emptyset)$ is defined and equal to c.

(ii) Let $\mathscr{B}_1$ and $\mathscr{B}_2$ be two blocks. If $\underline{\mathscr{B}}_1 \cap \underline{\mathscr{B}}_2 = \emptyset$ then the set of clauses $\{c_1 \cup c_2, c_1 \in \mathscr{B}_1, c_2 \in \mathscr{B}_2\}$ is called the *product* of $\mathscr{B}_1$ and $\mathscr{B}_2$ and is denoted by $\mathscr{B}_1 \cdot \mathscr{B}_2$. The set $\underline{\mathscr{B}}_1 \cup \underline{\mathscr{B}}_2$ support of every clause in the product is called the *support of* $\mathscr{B}_1 \cdot \mathscr{B}_2$. If a product is denoted by $\mathscr{P}$ then its support will be referred to as $\underline{\mathscr{P}}$. For instance, $\mathscr{B}^+(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^-(\{\underline{3},\underline{4}\}) = \{\{1,2,3,4'\},\{1,2,3',4\},\{1',2',3,4'\},\{1',2',3',4\}\}$.

Remark. For any block $\mathscr{B}$, the product $\mathscr{B}\cdot\mathscr{B}^{-}(\emptyset)$ is defined and equal to $\mathscr{B}$.

(iii) The products of n blocks or clauses are defined inductively.

Theorem 5.1 (Meta-resolvent Theorem). *Let $\mathscr{B}_1$ and $\mathscr{B}_2$ be blocks and let $k = |\underline{\mathscr{B}}_1 \cap \underline{\mathscr{B}}_2|$. If $k > 0$ then the meta-resolvent $\mathscr{B}_1 \bigtriangledown \mathscr{B}_2$ is the set of minimal clauses which may be derived from the clauses of $\mathscr{B}_1 \cup \mathscr{B}_2$, each of them being obtained with $2^k - 1$ elementary resolutions.*

Proof. If $\mathscr{B}_1 = \mathscr{B}_2$, the meta-resolvent is empty and no couple of clauses in $\mathscr{B}_1 \cup \mathscr{B}_2$ defines a resolvent; the result is trivial. Otherwise, we know by Comparison Theorem that any consequence from $\mathscr{B}_1 \cup \mathscr{B}_2$ contains an initial clause or a clause of their meta-resolvent. Conversely, we suppose that d belongs to $\mathscr{B}_1 \bigtriangledown \mathscr{B}_2$ and proceed by induction on k. We denote by ε_1 and ε_2 the parity of $\mathscr{B}_1$ and $\mathscr{B}_2$, respectively.

Basis. We suppose that $k = 1$ and $\{\underline{i}\} = \underline{\mathscr{B}}_1 \cap \underline{\mathscr{B}}_2$. We denote by d_1 (resp. d_2) the restriction of d to $\underline{\mathscr{B}}_1 \setminus \underline{\mathscr{B}}_2$ (resp. to $\underline{\mathscr{B}}_2 \setminus \underline{\mathscr{B}}_1$); d is the resolvent both of $d_1 \cup \{i\}$ and $d_2 \cup \{i'\}$ and of $d_1 \cup \{i'\}$ and $d_2 \cup \{i\}$. If $d_1 \cup \{i\} \in \mathscr{B}_1$ then $\operatorname{sgn}(d_2 \cup \{i'\}) = \operatorname{sgn}(d_1 \cup d_2)\cdot\operatorname{sgn}(d_1 \cup \{i'\}) = -\varepsilon_1 \cdot \varepsilon_2 \cdot -\varepsilon_1 = \varepsilon_2$ and $d_2 \cup \{i'\} \in \mathscr{B}_2$. Otherwise, $d_1 \cup \{i'\} \in \mathscr{B}_1$ and $d_2 \cup \{i\} \in \mathscr{B}_2$. So, each clause d of the meta-resolvent is derived from clauses of $\mathscr{B}_1 \cup \mathscr{B}_2$ with a single elementary resolution.

Induction step. We suppose that $k > 1$ and $\underline{i} \in \underline{a}_1 \cap \underline{a}_2$; we denote by $\underline{b}_1$ (resp. $\underline{b}_2$) the set $\underline{\mathscr{B}}_1 \setminus \{\underline{i}\}$ (resp. $\underline{\mathscr{B}}_2 \setminus \{\underline{i}\}$). As $\mathscr{B}_1 = \{i\}\cdot\mathscr{B}^{\varepsilon_1}(\underline{b}_1) \cup \{i'\}\cdot\mathscr{B}^{-\varepsilon_1}(\underline{b}_1)$, $\mathscr{B}_1 \cup \mathscr{B}_2$ may be written as $\{i\}\cdot[\mathscr{B}^{\varepsilon_1}(\underline{b}_1) \cup \mathscr{B}^{\varepsilon_2}(\underline{b}_2)] \cup \{i'\}\cdot[\mathscr{B}^{-\varepsilon_1}(\underline{b}_1) \cup \mathscr{B}^{-\varepsilon_2}(\underline{b}_2)]$.
As $\mathscr{B}^{\varepsilon_1}(\underline{b}_1) \bigtriangledown \mathscr{B}^{\varepsilon_2}(\underline{b}_2) = \mathscr{B}^{-\varepsilon_1}(\underline{b}_1) \bigtriangledown \mathscr{B}^{-\varepsilon_2}(\underline{b}_2) = \mathscr{B}_1 \bigtriangledown \mathscr{B}_2$, by the induction hypohesis we know that d can be derived from $\mathscr{B}^{\varepsilon_1}(\underline{b}_1) \cup \mathscr{B}^{\varepsilon_2}(\underline{b}_2)$ and from $\mathscr{B}^{-\varepsilon_1}(\underline{b}_1) \cup \mathscr{B}^{-\varepsilon_2}(\underline{b}_2)$ with $2^{k-1} - 1$ elementary resolutions. Consequently, $d \cup \{i\}$ and $d \cup \{i'\}$ can both be derived from $\mathscr{B}_1 \cup \mathscr{B}_2$ and d is obtained with $2\cdot(2^{k-1} - 1) + 1 = 2^k - 1$ elementary resolutions.

5.2. Replacement rules

Rule 1. *If $\underline{c}_1 \cap \underline{c}_2 \neq \emptyset$ then $(\mathscr{B}^{\varepsilon_1}(\underline{c}_1) \cup \mathscr{B}^{\varepsilon_2}(\underline{c}_2)) < \mathscr{B}^{-\varepsilon_1\cdot\varepsilon_2}(\underline{c}_1 \triangle \underline{c}_2)$* (Meta-resolvent Theorem).

Rule 2. *If $c_1 \cap c_2 = \emptyset$ then $(\mathscr{B}^{\varepsilon_1}(\underline{c}_1)\cdot\mathscr{B}^{\varepsilon_2}(\underline{c}_2) \cup \mathscr{B}^{-\varepsilon_1}(\underline{c}_1)\cdot\mathscr{B}^{-\varepsilon_2}(\underline{c}_2)) = \mathscr{B}^{\varepsilon_1\cdot\varepsilon_2}(\underline{c}_1 \cup \underline{c}_2)$.*

Proof. An even (resp. odd) number of negative literals on $\underline{c}_1 \cup \underline{c}_2$ is either the sum of an even number of negative literals on $\underline{c}_1$ and of an even (resp. odd) number of negative literals on $\underline{c}_2$, or the sum of an odd number of negative literals on $\underline{c}_1$ and of an odd (resp. even) number of negative literals on $\underline{c}_2$.

Remark. If $|\underline{c}_1| = |\underline{c}_2| = 1$, the two products are reduced to a single clause.

Rule 3. *Let $\mathscr{P}_1$, $\mathscr{P}_2$, $\mathscr{P}_3$ and $\mathscr{P}$ be products. If the supports $\underline{\mathscr{P}}_1$, $\underline{\mathscr{P}}_2$ and $\underline{\mathscr{P}}_3$ are disjoint from $\underline{\mathscr{P}}$ and if $\mathscr{P}_1 \cup \mathscr{P}_2 < \mathscr{P}_3$ then the products $\mathscr{P}_1 \cdot \mathscr{P}$, $\mathscr{P}_2 \cdot \mathscr{P}$ and $\mathscr{P}_3 \cdot \mathscr{P}$ are defined and*

$$(\mathscr{P}_1 \cdot \mathscr{P} \cup \mathscr{P}_2 \cdot \mathscr{P}) < \mathscr{P}_3 \cdot \mathscr{P}.$$

Proof. As $\underline{\mathscr{P}}_1$, $\underline{\mathscr{P}}_2$ and $\underline{\mathscr{P}}_3$ are disjoint from $\underline{\mathscr{P}}$ the products are defined. Moreover, a solution of $\mathscr{P}_3 \cdot \mathscr{P}$ is either a solution of $\mathscr{P}$ or a solution of $\mathscr{P}_3$ hence a solution of $\mathscr{P}_1 \cup \mathscr{P}_2$. In both cases, that solution satisfies $\mathscr{P}_1 \cdot \mathscr{P} \cup \mathscr{P}_2 \cdot \mathscr{P}$.

Examples.

$$\{3\} \cdot \mathscr{B}^+(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}) \cup \{3'\} \cdot \mathscr{B}^-(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}) < \mathscr{B}^+(\{\underline{1},\underline{2},\underline{3}\}) \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}).$$

$$\{1,3\} \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}) \cup \{2',3'\} \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}) < \{1,2'\} \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}).$$

$$\mathscr{B}^+(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}) \cup \mathscr{B}^-(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}) < \mathscr{B}^+(\{\underline{4},\underline{5}\}).$$

Rule 4. *Let c_1, c_2 be clauses and $\mathscr{P}_1, \mathscr{P}_2$ be products with disjoint supports. If c_1 and c_2 define a resolvent and if their supports $\underline{c}_1$ and $\underline{c}_2$ are disjoint from $\underline{\mathscr{P}}_1 \cup \underline{\mathscr{P}}_2$ then the products $c_1 \cdot \mathscr{P}_1$, $c_2 \cdot \mathscr{P}_2$ and $(c_1 \triangledown c_2) \cdot \mathscr{P}_1 \cdot \mathscr{P}_2$ are defined and*

$$(c_1 \cdot \mathscr{P}_1 \cup c_2 \cdot \mathscr{P}_2) < (c_1 \triangledown c_2) \cdot \mathscr{P}_1 \cdot \mathscr{P}_2.$$

Proof. As $\underline{c}_1$ and $\underline{c}_2$ are disjoint from $\underline{\mathscr{P}}_1 \cup \underline{\mathscr{P}}_2$ the products are defined. Moreover, for any couple $(d_1, d_2) \in \mathscr{P}_1 \times \mathscr{P}_2$, the clauses $c_1 \cup d_1$ and $c_2 \cup d_2$ define a resolvent equal to $(c_1 \bigtriangledown c_2) \cup d_1 \cup d_2$. For instance, $\{3\} \cdot \mathscr{B}^+(\{\underline{1},\underline{2}\}) \cup \{3'\} \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\}) < \mathscr{B}^+(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^+(\{\underline{4},\underline{5}\})$.

Rule 5. Let $\mathscr{B}_1, \mathscr{B}_2$ be blocks and $\mathscr{P}_1, \mathscr{P}_2$ be products with disjoint supports. If $\underline{\mathscr{B}}_1 \cap \underline{\mathscr{B}}_2 \neq \emptyset$ and if $\underline{\mathscr{B}}_1 \cup \underline{\mathscr{B}}_2$ is disjoint from $\underline{\mathscr{P}}_1 \cup \underline{\mathscr{P}}_2$ then the products $\mathscr{B}_1 \cdot \mathscr{P}_1$, $\mathscr{B}_2 \cdot \mathscr{P}_2$ and $(\mathscr{B}_1 \bigtriangledown \mathscr{B}_2) \cdot \mathscr{P}_1 \cdot \mathscr{P}_2$ are defined and

$$(\mathscr{B}_1 \cdot \mathscr{P}_1 \cup \mathscr{B}_2 \cdot \mathscr{P}_2) < (\mathscr{B}_1 \bigtriangledown \mathscr{B}_2) \cdot \mathscr{P}_1 \cdot \mathscr{P}_2.$$

Proof. The products are clearly defined. As $\underline{\mathscr{B}}_1 \cap \underline{\mathscr{B}}_2 \neq \emptyset$, we conclude by Meta-resolvent Theorem that any clause of $(\mathscr{B}_1 \bigtriangledown \mathscr{B}_2) \cdot \mathscr{P}_1 \cdot \mathscr{P}_2$ may be derived by elementary resolutions from clauses of $\mathscr{B}_1 \cdot \mathscr{P}_1 \cup \mathscr{B}_2 \cdot \mathscr{P}_2$. For instance, $(\mathscr{B}^+(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^+(\{\underline{3},\underline{4}\} \cup \mathscr{B}^-(\{\underline{1},\underline{2}\}) \cdot \mathscr{B}^+(\{\underline{5},\underline{6}\}) < \mathscr{B}^+(\{\underline{3},\underline{4}\}) \cdot \mathscr{B}^+(\{\underline{5},\underline{6}\})$.

6. Meta-resolution trees

We now define a representation of formulae by ternary trees; this representation will enable us to detect blocks and apply rules. In classical resolution trees, the many

occurrences of the same clause may lead to an exponential size; opposite to that case, our new data structure is computable in linear time with respect to the size of the input formula.

6.1. *Tree representing a formula*

(i) *Ordering.* An order on the set $\underline{A}$ of atoms being given, this order is extended to the union $\mathscr{A}$ of atoms and literals by the convention: $\underline{i} < j \Rightarrow i < \underline{i} < i' < j < \underline{j} < j'$.

(ii) *Word representation of a clause.* Let c be a clause and $\underline{k}$ be the greatest element of $\underline{c}$ with respect to the given order. The word $\mathrm{wd}(c)$ of $\mathscr{A}^*$ representing c is the word of length k whose ith letter is $\underline{i}$ if it does not belong to c and otherwise the literal i or i' which belongs to c. For instance, the given order being the natural one, the word representing $\{2,4,5'\}$ is $\{\underline{1},2,\underline{3},4,5'\}$.

(iii) *Normalized formula.* Let F be a formula and let $F_0 = \{c \in F,\ \exists d \in F,\ c \supset d,\ d \neq c\}$. The normalized formula of F is $\mathrm{NF} = F \setminus F_0$. We can remark that F and NF have the same set of solutions.

Definition 6.1. Let F be a formula. The *tree representing* F is the lexical tree (S, E) of the set of words wd(NF). This tree may be seen as an automaton whose initial state is the root and terminal states are the leaves, each one recognizing a word $\mathrm{wd}(c)$. Each state s has at most three sons; a son t of s is called the *left, right or medium son* if the edge (s,t) is indexed respectively by a positive literal, a negative one or an atom. An edge from heights $i-1$ to i is indexed by i, i' or $\underline{i}$.

For instance if $F = \{\{1,2\},\{1,2,3\},\{1',2\},\{2',3\},\{2',3'\}\}$ then $\mathrm{NF} = \{\{1,2\},\{1',2\},\{2',3\},\{2',3'\}\}$, $\mathrm{wd(NF)} = \{\{1,2\},\{1',2\},\{\underline{1},2',3\},\{\underline{1},2',3'\}\}$ and the tree associated to F is given by Fig. 5.

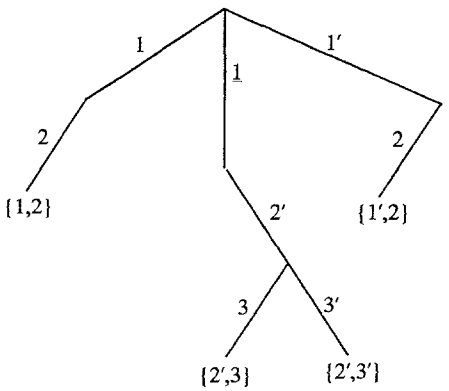


Fig. 5. The tree representing $F = \{\{1,2\},\{1,2,3\},\{1',2\},\{2',3\},\{2',3'\}\}$.

Remark. The clauses of $F \setminus \mathrm{NF}$ can easily be detected and suppressed during the construction of the tree.

6.2. *Adding products*

(i) *Data structure.* A non-empty supported product $\mathscr{P}$ on A will be represented by an $\underline{A}$-indexed array with the following conventions:

- If $\mathscr{B}^{\varepsilon}(\{\underline{i}_1, \underline{i}_2, \ldots, \underline{i}_k\})$ is a factor of $\mathscr{P}$ then for $j = 1, \ldots, k-1$ the i_jth element of the array is $\underline{i}_{j+1}$ and the i_kth is ε.
- If $c = \{\ell_1, \ell_2, \ldots, \ell_r\}$ is a factor of $\mathscr{P}$ then for $j = 1, \ldots, r$, the l_jth element of the array is $+$ (resp, $-$) if ℓ_j is a positive (resp. negative) literal.
- If $\underline{i}$ does not belong to $\underline{\mathscr{P}}$ then the ith element of the array is 0.

For instance, if $\underline{A} = \{\underline{1}, \ldots, \underline{9}\}$ the product $\{1, 4, 9'\} \cdot \mathscr{B}^{+}(\{\underline{2}, \underline{5}, \underline{7}\}) \cdot \mathscr{B}^{-}(\{\underline{6}, \underline{8}\})$ will be represented by $[+, \underline{5}, 0, +, \underline{7}, \underline{8}, +, -, -]$.

Remark. In the following figures, the previous notation of products that are easier to understand will be used.

(ii) *Adding products procedure.* The AP procedure receives two products, each represented by an array. The rules 1 to 5 introduced in Section 5.2 are tested and applied if possible. In that case the output consists of a single product and otherwie of the union of the two input products.

(iii) *Adding unions of products.* Let $\mathscr{P}, \mathscr{P}_1, \ldots, \mathscr{P}_k$ be products. The recursive procedure RAP is defined as follows:

$$\mathrm{RAP}(\mathscr{P}_1, \mathscr{P}) = \mathrm{AP}(\mathscr{P}_1, \mathscr{P}).$$

$$\mathrm{RAP}(\mathscr{P}_1 \cup \cdots \cup \mathscr{P}_k, \mathscr{P}) = \begin{cases} \mathrm{RAP}(\mathscr{P}_2 \cup \cdots \cup \mathscr{P}_k, \mathscr{Q}) & \text{if } \mathrm{AP}(\mathscr{P}_1, \mathscr{P}) \text{ consists of a single product } \mathscr{Q} \\ \mathscr{P}_1 \cup \mathrm{RAP}(\mathscr{P}_2 \cup \cdots \cup \mathscr{P}_k, \mathscr{P}) & \text{otherwise.} \end{cases}$$

Let $\mathscr{U}_1$ be a union of products. The procedure AU is defined as follows:

$$\mathrm{AU}(\mathscr{U}_1, \emptyset) = \mathscr{U}_1; \qquad \mathrm{AU}(\mathscr{U}_1, \mathscr{P}_1) = \mathrm{RAP}(\mathscr{U}_1, \mathscr{P}_1).$$

$$\mathrm{AU}(\mathscr{U}_1, \mathscr{P}_1 \cup \cdots \cup \mathscr{P}_k) = \begin{cases} \mathrm{AU}(\mathscr{U}_1, \mathscr{P}_2 \cup \cdots \cup \mathscr{P}_k) \cup \mathscr{P}_1 & \text{if } \mathrm{RAP}(\mathscr{U}_1, \mathscr{P}_1) = \mathscr{U}_1 \cup \mathscr{P}_1, \\ \mathrm{AU}(\mathrm{RAP}(\mathscr{U}_1, \mathscr{P}_1), \mathscr{P}_2 \cup \cdots \cup \mathscr{P}_k) & \text{otherwise.} \end{cases}$$

6.3. *Meta-resolution algorithm*

Union attached to a state. Let F be a formula and (S, E) be the tree representing F. The union $\phi(s)$ of blocks attached to a state s is defined recursively as follows:

If s is a leaf then $\phi(s)$ is empty.

Otherwise, let i be the height of any son of s and $\phi(s)$ be initialized to $\emptyset$:
If s has a left son $\mathrm{ls}(s)$ then $\phi(s)$ is replaced by $\mathrm{AU}(i\cdot\phi(\mathrm{ls}(s)),\phi(s))$.
If s has a right son $\mathrm{rs}(s)$ then $\phi(s)$ is replaced by $\mathrm{AU}(i'\cdot\phi(\mathrm{ls}(s)),\phi(s))$.
If s has a middle son $\mathrm{ms}(s)$ then $\phi(s)$ is replaced by $\mathrm{AU}(\phi(\mathrm{ms}(s)),\phi(s))$.

Definition 6.2. With the previous notations, (S,E,ϕ) is called a *meta-resolution tree* of F.

Lemma 6.3 (Complexity Lemma). *Let n be the number of atoms and m be the number of clauses of F. A meta-resolution tree of F is computed in time $O(n\cdot m^2)$.*

Proof. The construction of (S,E) requires a time in $O(n)$ for each clause and a total time in $O(n\cdot m)$. The number of calls to AU is at most m and the maximal number of blocks in a union is m. As AU is designed in such a way that two different blocks are submitted only once to AP, the total number of calls to AP is in $O(m^2)$. Each call to AP requiring a time in $O(n)$, the computation of (S,E,ϕ) is in time $O(n\cdot m^2)$.

An example is given in Fig. 6.

Lemma 6.4 (Products Recognition Lemma). *Let $\mathscr{P}$ be a block. The union at the root of a meta-resolution of $\mathscr{P}$ is $\mathscr{P}$ itself.*

Proof. We proceed by induction on $|\underline{\mathscr{P}}|$.

Basis. If $|\underline{\mathscr{P}}|=1$, $\mathscr{P}$ is reduced to a single clause of length 1 and the result is trivial.

Induction step. Let $\underline{i}$ be the smallest atom in $\mathscr{P}$ and $\mathscr{E}$ be the factor of $\mathscr{P}$ in which $\underline{i}$ appears ($\mathscr{E}$ is either a clause or a block). We denote by $\mathscr{Q}$ the only product such that $\mathscr{P}=\mathscr{E}\mathscr{Q}$.

The states of height 0 to $i-1$ have only one son which is a medium one and so the union attached to the root is equal to the union attached to the only state s of height $i-1$.

If $\mathscr{E}$ is a clause c containing i (resp. i') then s has only one son, the left one (resp. right one), and $\phi(s)=\{i\}\cdot\phi(\mathrm{ls}(s))$ (resp. $\{i'\}\cdot\phi(\mathrm{rs}(s))$). By the induction hypothesis, $\phi(\mathrm{ls}(s))=(c\setminus\{i\})\mathscr{Q}$ (resp. $(c\setminus\{i'\})\mathscr{Q}$ and $\phi(s)$ is equal to $\mathscr{P}$.

If $\mathscr{E}$ is a block $\mathscr{B}$ then s has two sons, the right and left ones, and then $\phi(s)=\mathrm{AU}(\{i\}\cdot\phi(\mathrm{ls}(s)),\{i'\}\cdot\phi(\mathrm{rs}(s)))$. Let ε denote the sign of $\mathscr{B}$. By the induction hypothesis, $\phi(\mathrm{ls}(s))=\mathscr{B}^{\varepsilon}(\underline{\mathscr{B}}\setminus\{\underline{i}\})\mathscr{Q}$ and $\phi(\mathrm{rs}(s))=\mathscr{B}^{-\varepsilon}(\underline{\mathscr{B}}\setminus\{\underline{i}\})\mathscr{Q}$.

By rule 3, $\{i\}\cdot\mathscr{B}^{\varepsilon}(\underline{\mathscr{B}}\setminus\{\underline{i}\})\mathscr{Q}\cup\{i'\}\cdot\mathscr{B}^{\varepsilon}(\underline{\mathscr{B}}\setminus\{\underline{i}\})\mathscr{Q}<(\{i\}\cdot\mathscr{B}^{\varepsilon}(\underline{\mathscr{B}}\setminus\{\underline{i}\})\cup\{\underline{i}'\}\cdot\mathscr{B}^{\varepsilon}(\mathscr{B}\setminus\{i\}))\mathscr{Q}$ which, by rule 2, is equal to $\mathscr{B}\mathscr{Q}=\mathscr{P}$.

So, in both cases, the union attached to s is $\mathscr{P}$ and the proof is complete.

Theorem 6.5 (Contradiction Theorem). *If the contradictory block $\mathscr{B}^{+}(\emptyset)$ is obtained at the root of a meta-resolution tree then the input formula is contradictory.*

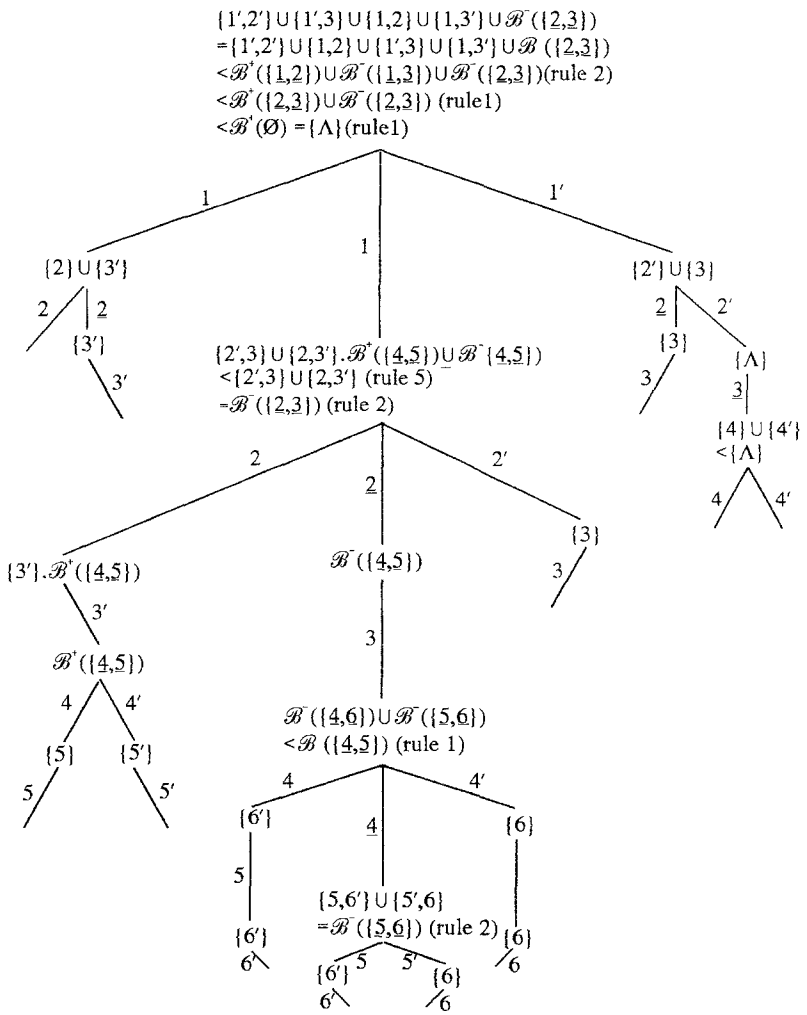


Fig. 6. A meta-resolution tree of the formula $F = \{\{1,2\},\{1',2',4\},\{1',2',4'\},\{1,3'\},\{1',3\},\{2,3',4,5\},\{2,3',4',5'\},\{2',3\},\{4,6'\},\{4',6\},\{5,6'\},\{5',6\}\}$.

Proof. As the union at the root is obtained by applying recursively the replacement rules, its set of solutions contains the one of the input formula. If this expression is the contradictory block then the two sets of solutions are empty and the input formula is contradictory.

Remark. The converse is not true and meta-resolution is unable to detect some contradictions. We shall now prove that it detects Tseitin's contradictory formulae efficiently. Of course, as shown by Fig. 6, it may recognize more general contradictions.

7. Tseitin's contradictory formulae

In this part, $G = (X, U, \sigma)$ is an alternated graph on $\underline{A}$ containing an odd connected component.

7.1. The $G(x)$ reduction lemmas

Lemma 7.1. *Let x be a vertex of G. The union attached to the root of a tree of $G(x)$ is $\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))$.*

Proof. We proceed by induction $|\mathrm{fi}(x)|$.

Basis. If $|\mathrm{fi}(x)| = 0$, the result is given by Lemma 6.4.

Induction step. Let $\underline{i}$ be the smallest atom in $\mathrm{fi}(x)$ and d be a subclause recognized by a state of height $i-1$ in the tree representing $G(x)$. The state has only two sons, the right and the left ones which are at the root of subtrees encoding the same formula:

$$\{c_1 \cup c_2,\ c_1 \in \mathscr{B}^{-\sigma(x)\cdot \mathrm{sgn}(d)}(\mathrm{fe}(x)\setminus\{\underline{d}\}), \underline{c}_2 = \mathrm{fi}(x)\setminus\{\underline{i}\}\}.$$

By the induction hypothesis, the union attached to the state of height $i-1$ is

$$\{i\}\cdot\mathscr{B}^{-\sigma(x)\cdot\mathrm{sgn}(d)}(\mathrm{fe}(x)\setminus\{\underline{d}\})\cup\{i'\}\cdot\mathscr{B}^{-\sigma(x)\cdot\mathrm{sgn}(d)}(\mathrm{fe}(x)\setminus\{\underline{d}\})$$

$$< \mathscr{B}^{-\sigma(x)\cdot\mathrm{sgn}(d)}(\mathrm{fe}(x)\setminus\{\underline{d}\}) \quad \text{by rule 3.}$$

So, the indices of loops are eliminated and the union attached to the root is $\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))$.

Lemma 7.2. *Let x and y be two vertices of G. If x and y are neighbours then a meta-resolution of $G(x)\cup G(y)$ gives $\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))$.*

Proof. Let i be the smallest index in $f(x)\cup f(y)$. The tree representing $G(x)\cup G(y)$ has only one state s of height $i-1$. The atom $\underline{i}$ is either the index of an edge containing both x and y or the index of a loop containing only one of these two vertices.

We first suppose that, for instance, the edge $\underline{i}$ is a loop on x. The path associated to a clause of $G(x)$ (resp. $G(y)$) contains either the right son or the left son of s (resp. always the middle son of s). By Lemma 7.1, the union attached to s is $\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))\cup\mathscr{B}^{-\sigma(y)}(\mathrm{fe}(y))$. As $\mathrm{fe}(x)$ and $\mathrm{fe}(y)$ are not disjoint, rule 1 applies and the union obtained is $\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))$.

If the edge i contains x and y then s has no middle son. We proceed by induction on the number of edges containing both x and y.

Basis. By Lemma 7.1, the union attached to the right son (resp. left son) of s is $\mathscr{B}^{\sigma(x)}(\mathrm{fe}(x)\setminus\{\underline{i}\})\cup\mathscr{B}^{\sigma(y)}(\mathrm{fe}(y)\setminus\{\underline{i}\})$ (resp. $\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x)\setminus\{\underline{i}\})\cup\mathscr{B}^{-\sigma(y)}(\mathrm{fe}(y)\setminus\{\underline{i}\})$. So,

the union attached to s is

$$\begin{aligned}
&\{i'\}\cdot[\mathscr{B}^{\sigma(x)}(\mathrm{fe}(x)\setminus\{\underline{i}\})\cup\mathscr{B}^{\sigma(y)}(\mathrm{fe}(y)\setminus\{\underline{i}\})]\\
&\quad\cup\{i\}\cdot[\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x)\setminus\{\underline{i}\})\cup\mathscr{B}^{-\sigma(y)}(\mathrm{fe}(y)\setminus\{\underline{i}\})]\\
&=\{i'\}\cdot[\mathscr{B}^{\sigma(x)}(\mathrm{fe}(x)\setminus\{\underline{i}\})]\cup\{i\}\cdot[\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x)\setminus\{\underline{i}\})]\\
&\quad\cup\{i'\}\cdot[\mathscr{B}^{\sigma(y)}(\mathrm{fe}(y)\setminus\{\underline{i}\})]\cup\{i\}\cdot[\mathscr{B}^{-\sigma(y)}(\mathrm{fe}(y)\setminus\{\underline{i}\})]\\
&<\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))\cup\mathscr{B}^{-\sigma(y)}(\mathrm{fe}(y))\quad\text{by rule 2}\\
&<\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))\quad\text{by rule 1.}
\end{aligned}$$

Induction step. By the induction hypothesis, the union attached to the right son (resp. left son) of s is $\mathscr{B}^{-(-\sigma(x)\cdot-\sigma(y))}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))$ (resp. $\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))$). So the union attached to s is

$$\begin{aligned}
&\{i\}\cdot\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))\cup\{i'\}\cdot\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))\\
&\quad<\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x)\triangle\mathrm{fe}(y))\quad\text{by rule 3.}
\end{aligned}$$

An example is given by Figs. 7 and 8.

Property 7.3. Let i be an edge of G with two ends x and y. We consider the map τ from X to itself defined by $\tau(y_0)=x_0$ and $\forall x\neq y_0$, $\tau(x)=x$; the map τ can also be considered as a map on U defined by $\tau(\{x,y\})=\{\tau(x),\tau(y)\}$. Let $\tau\sigma$ be the sign defined on $\tau(X)$ by $\tau\sigma(x_0)=\sigma(x_0)\cdot\sigma(y_0)$ and $\forall x\neq x_0$, $\tau\sigma(x)=\sigma(x)$. Under those conditions, $\mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x_0)\triangle\mathrm{fe}(y_0))=\mathscr{B}^{-\tau\sigma(x_0)}(\mathrm{fe}(\tau(x_0))$ and that block is obtained directly when G is replaced by $H=(\tau(X),\tau(U),\tau\sigma)$. (See Figs. 9 and 10).

Figures. Let for instance x and y be given by Fig. 7:

$$G(x)=\{\{1,2,4\},\{1,2',4\},\{1',2,4'\},\{1',2',4'\}\}\quad\text{and}$$

$$G(y)=\{\{1,3,5'\},\{1,3',5\},\{1',3,5\},\{1',3',5'\}\}$$

Fig. 8 shows a meta-resolution tree of $G(x)\cup G(y)$.

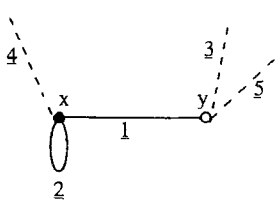


Fig. 7. Two vertices of G.

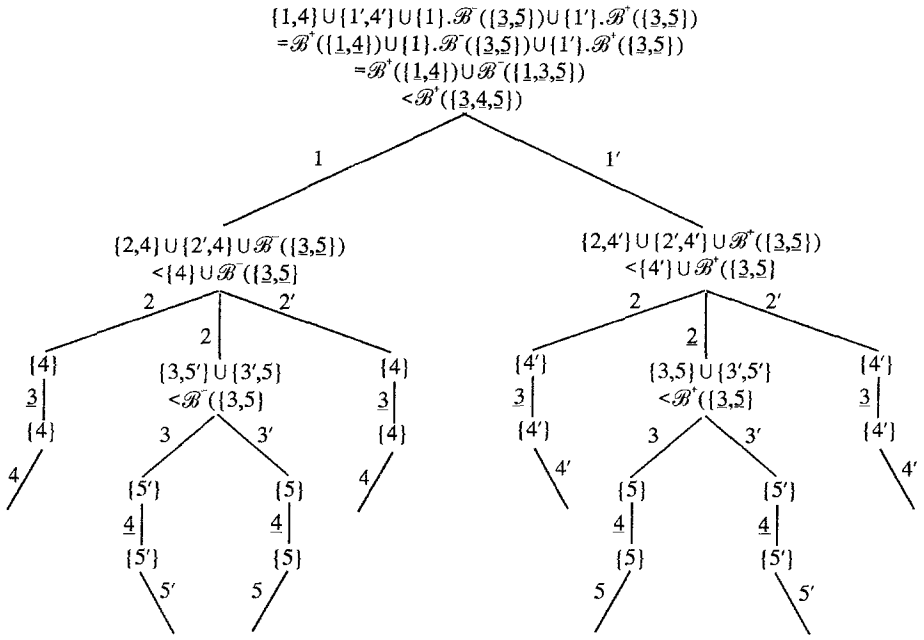


Fig. 8. A meta-resolution of $G(x) \cup G(y)$.

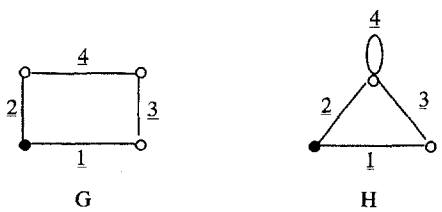


Fig. 9. A graph G and a graph H obtained from G by τ.

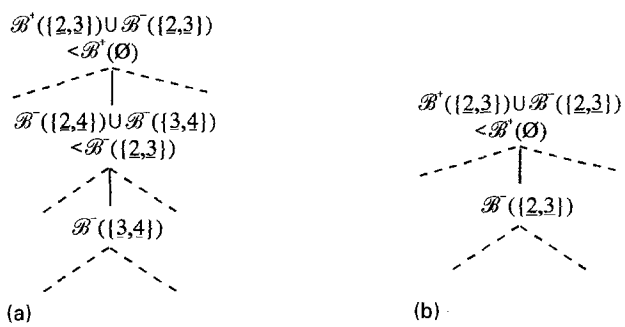


Fig. 10. (a) Meta-resolution of $C(G)$. (b) Meta-resolution of $C(H)$.

As shown in Fig. 10, the block $\mathscr{B}^-(\{\underline{2},\underline{3}\})$ is derived from two blocks in the tree of $C(G)$ and obtained directly in the tree of $C(H)$.

Theorem 7.4 (Odd graph Theorem). *A meta-resolution tree of $C(G)$ gives the contradictory block in time* $O(|U||C(G)|^2)$.

Proof. We first prove that the contradictory block is obtained. The subformulae defined by two disjoint components have disjoint supports. So, we restrict ourselves to the case of an odd connected alternated graph G. By its recursive definition and the $G(x)$ reduction lemmae, the union attached to the root of a tree representing $C(G)$ is $\mathscr{B}_1 \cup \mathscr{B}_2 \cup \cdots \cup \mathscr{B}_k$ where each $\mathscr{B}_i$ $(i = 1, \ldots, k)$ is either $\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))$ for some vertex x of G or $\mathscr{B}^{-\sigma(y)\cdot\sigma(z)}(\mathrm{fe}(y) \triangle \mathrm{fe}(z))$ for some neighbour vertices y, z of G. We proceed by induction on the number k of blocks.

Basis. If $k = 1$ there are two possible cases:

- G has only one vertex x. In that case $\sigma(x) = -$, $\mathrm{fe}(x) = \emptyset$ and $\mathscr{B}_1 = \mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x)) = \mathscr{B}^+(\emptyset)$.
- G has two vertices x and y. In that case $\sigma(x)\cdot\sigma(y) = -$, $\mathrm{fe}(x) \triangle \mathrm{fe}(y) = \emptyset$ and $\mathscr{B}_1 = \mathscr{B}^{-\sigma(x)\cdot\sigma(y)}(\mathrm{fe}(x) \triangle \mathrm{fe}(y)) = \mathscr{B}^+(\emptyset)$.

Induction step. As each edge contains two vertices, during the computation rule 1 will be applied to two blocks $\mathscr{B}_i\mathscr{B}_j$ with non-disjoint supports. So the number of blocks is reduced to $k - 1$ and we conclude by the induction hypothesis together with the property 7.3.

This step is illustrated by Figs. 9 and 10. So, a meta-resolution tree recognizes a contradictory formula of Tseitin.

Let n denote $|A|$ and m denote $|C(G)|$. By Complexity Lemma the computation of a meta-resolution tree of $C(G)$ requires a time in $O(n\cdot m^2)$.

Remark. Using Margulis' graphs [11], Galil [8] has given examples where the maximal length of an expression at a state grows linearly with n. In that case, the complexity of resolution grows exponentially (see [7, 14]).

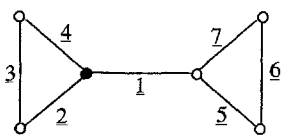
Corollary 7.5 (Minimal Subformula Corollary). *Meta-resolution applied to the minimal contradictory subformula of $C(G)$ gives the contradictory block in time* $O(|U||C(G)|^2)$.

Proof. We restrict ourselves to the case of an odd connected alternated graph G. By the inclusion theorem, we know that the minimal contradictory subformula of $C(G)$ is its subset of simple clauses. Let x be a vertex of G and let $H_0 = (\{x_0\}, \emptyset)$, $H_1, H_2, \ldots, H_k$ be the connected components of $G \setminus f(x)$. We denote by f_i $(i = 1, \ldots, k)$ the subset of edges in $\mathrm{fe}(x)$ connecting x with H_i. Let c be a simple clause of $G(x)$ and let c_i be the restriction of c to f_i. By the inclusion theorem, we know that H_0 is the only odd component of $G \setminus c$. Consequently, $\mathrm{sgn}(c_i) = \sigma(H_i)$ and the restriction of c to $\mathrm{fe}(x)$

belongs to the product $\mathscr{B}^{\sigma(H_1)}(f_1)\cdot\mathscr{B}^{\sigma(H_2)}(f_2)\cdots\cdot\mathscr{B}^{\sigma(H_k)}(f_k)$ which is included in $\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))$,

So, when $C(G)$ is replaced by its minimal contradictory subformula, the blocks are replaced by products of blocks with disjoint supports. We denote by $G(x)_s$ the subset of simple clauses of $G(x)$.

The proof of Lemma 7.1 applies to $G(x)_s$. We generalize the proof of Lemma 7.2: Let x and y be two neighbours and let i be the smallest index in $f(x)\cup f(y)$. The block $\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))$ (resp. $\mathscr{B}^{-\sigma(y)}(\mathrm{fe}(y))$) is replaced by $\mathscr{B}^{\sigma_1}(f_1)\cdots\cdot\mathscr{B}^{\sigma_k}(f_k)$ (resp.



A graph G

{{1,2,4},{1,2',4'},{1',5,7},{1',5',7'},{2,3'},{2',3}
{3,4'},{3',4},{5,6'},{5',6},{6,7'},{6',7}}

The minimal contradictory subformula of C(G)

Fig. 11. A graph G such that $C(G)$ is distinct from its minimal contradictory subformula.

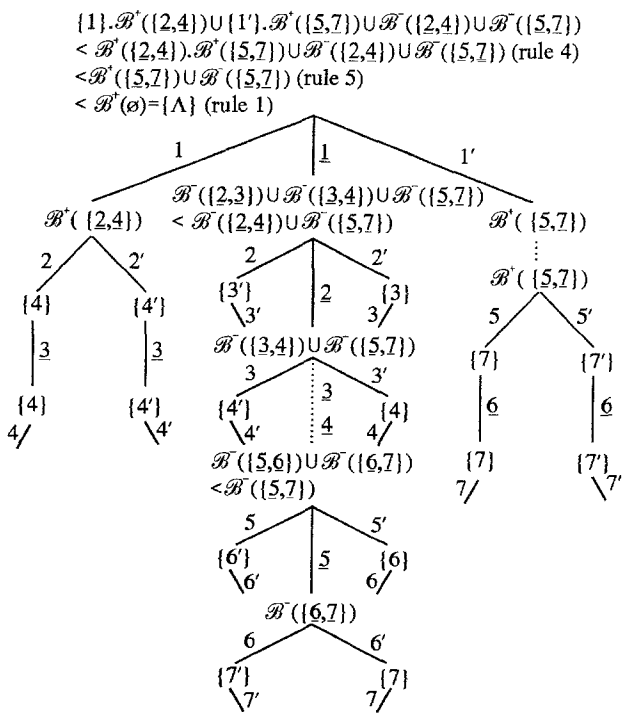


Fig. 12. A meta-resolution of the minimal contradictory subformula of $C(G)$.

$\mathscr{B}^{\varepsilon_1}(g_1)\cdots\cdot\mathscr{B}^{\varepsilon_l}(g_l)$). The vertex x (resp. y) belongs to exactly one component of $G\backslash f(y)$, g_1 for instance (resp. $G\backslash f(x)$, f_1 for instance).

If i contains only x (or y), meta-resolution gives, by rule 6,

$$\mathscr{B}^{-\sigma_1\cdot\varepsilon_1}(f_1\triangle g_1)\cdot\mathscr{B}^{\sigma_2}(f_2)\cdots\cdot\mathscr{B}^{\sigma_k}(f_k)\cdot\mathscr{B}^{\varepsilon_2}(g_2)\cdots\cdot\mathscr{B}^{\varepsilon_1}(g_1).$$

If the edge $\underline{i}$ contains x and y, we first suppose that $\underline{i}$ is the only edge between x and y. The expression obtained is $\{i\}\cdot(\mathscr{B}^{\sigma_1}(f_1\backslash\{\underline{i}\})\cdot\mathscr{P}_1\cup\mathscr{B}^{\varepsilon_1}(g_1\backslash\{\underline{i}\})\cdot\mathscr{P}_2)\cup\{i'\}\cdot(\mathscr{B}^{-\sigma_1}(f_1\backslash\{\underline{i}\})\cdot\mathscr{P}_1\cup\mathscr{B}^{-\varepsilon_1}(g_1\backslash\{\underline{i}\})\cdot\mathscr{P}_2)$ (where $\mathscr{P}_1=\mathscr{B}^{\sigma_2}(f_2)\cdots\cdot\mathscr{B}^{\sigma_k}(f_k)$ and $\mathscr{P}_2=\mathscr{B}^{\varepsilon_2}(g_2)\cdots\cdot\mathscr{B}^{\varepsilon_1}(g_1)$) which may be written as

$$\begin{aligned}&\mathscr{P}_1\cdot[\{i\}\cdot\mathscr{B}^{\sigma_1}(f_1\backslash\{\underline{i}\})\cup\{i'\}\cdot\mathscr{B}^{-\sigma_1}(f_1\backslash\{\underline{i}\})]\\&\quad\cup\mathscr{P}_2\cdot[\{i\}\cdot\mathscr{B}^{\varepsilon_1}(g_1\backslash\{\underline{i}\})\cdot\cup\{i'\}\cdot\mathscr{B}^{-\varepsilon_1}(g_1\backslash\{\underline{i}\})]\\&<\mathscr{P}_1\cdot\mathscr{B}^{\sigma_1}(f_1)\cup\mathscr{P}_2\cdot\mathscr{B}^{\varepsilon_1}(g_1)\quad\text{by rule 2}\\&<\mathscr{P}_1\cdot\mathscr{P}_2\cdot\mathscr{B}^{-\sigma_1\cdot\varepsilon_1}(f_1\triangle g_1)\quad\text{by rule 6.}\end{aligned}$$

The induction step is easily generalized and meta-resolution applied to $G(x)_s\cup G(y)_s$ gives $\mathscr{P}_1\cdot\mathscr{P}_2\cdot\mathscr{B}^{-\sigma_1\cdot\varepsilon_1}(f_1\triangle g_1)$. Using this result, the proof of Theorem 7.4 applies with few modifications. Figs. 11 and 12 give an example:

8. A general algorithm

8.1. The Davis–Putnam–Loveland reduction

Definition 8.1. Let $\underline{i}$ be an atom of $\underline{A}$, F be a formula and $F\underline{i}$ denote the subformula $\{c\in F, \underline{i}\notin\underline{c}\}$. The *reduction* of F by the true (resp. false) assignment for $\underline{i}$ is the formula $F\underline{i}\cup\{c, c\cup\{i'\}\in F\}$ (resp. $F\underline{i}\cup\{c, c\cup\{i\}\in F\}$) and is denoted by $Ri(F)$ (resp. $Ri'(F)$). For instance, if $F=\{\{1,2\},\{1',3\},\{2,3\}\}$ then $R1(F)=\{\{3\},\{2,3\}\}$ and $R1'(F)=\{\{2\},\{2,3\}\}$.

Property 8.2. *A valuation v such that $i\in v$ (resp. $i'\in v$) is a solution of F if and only if the restriction of v to $\underline{A}\backslash\{\underline{i}\}$ is a solution of $Ri(F)$ (resp. $Ri'(F)$).*

Proof. Let v be a solution of F and c be a clause of $Ri(F)$. If $c\in F\underline{i}$ then $c\in F$ and the restriction of v satisfies c. If $c\cup\{i'\}\in F$ then v satisfying $c\cup\{i'\}$ satisfies c and its restriction also satisfies c.

Conversely, let v be a valuation whose restriction satisfies $Ri(F)$ and such that $i\in v$ and let c be a clause of F. If $i'\in c$ then $c\backslash\{i'\}\in Ri(F)$ and is satisfied by the restriction of v; if $\underline{i}\notin\underline{c}$ then $c\in F\underline{i}$ and is satisfied by the restriction of v. If $i\in c$ then c is satisfied by v. So, in the three cases, c is satisfied by v.

The Davis–Putnam and Loveland procedure. The classical Davis–Putnam and Loveland (DPL) procedure [10] extends recursively a partial valuation by a true

literal and consequently reduces the formula. Either the formula reduces to the empty set and a solution is found or an empty clause appears. In the second case, the extension by the false literal is tried and either a solution is found or the contradiction of the formula is proved.

Algorithm 8.3. The algorithm integrates meta-resolution in the DPL procedure. It is obtained by a call to the recursive evaluate procedure which is:

Construct a meta-resolution tree.
If the contradictory block is not obtained and if F is not empty then
  evaluate ($Ri(F)$) (i being the smallest atom appearing in F).
  If no solution is found then
    evaluate ($Ri'(F)$).
    If once more no solution is found then F is contradictory.

Theorem 8.4. *The general algorithm either gives a solution of the input formula or proves its contradiction.*

Proof. We proceed by induction on $|\underline{A}|$.

Basis. If $|\underline{A}| = 0$ then either F is empty hence satisfiable or equal to the contradictory block.

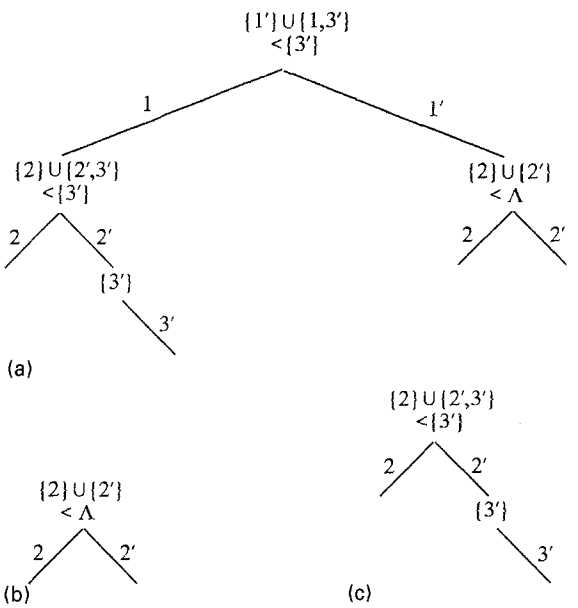


Fig. 13.

Induction step. If F is contradictory then by Property 8.2 both $Ri(F)$ and $Ri'(F)$ are contradictory. In that case, if meta-resolution fails we conclude by the induction hypothesis.

If F is satisfiable then by Theorem 6.5 the contradictory block is not obtained by meta-resolution. In that case, by Property 8.2 at least one of the two formulae $Ri(F)$ and $Ri'(F)$ is satisfiable and we conclude by the induction hypothesis.

Example. Let $F = \{\{1,2\},\{1,2',3'\},\{1',2\},\{1',2'\}\}$. The algorithm first constructs the tree given by Fig. 13(a). As the contradictory block is not obtained, the procedure evaluate is called on $R1(F) = \{\{2\},\{2'\}\}$ and proves its contradiction by meta-resolution (Fig. 13(b)). So, a second call is made on $R1'(F) = \{\{2\},\{2',3'\}\}$ and a new tree constructed (Fig. 13(c)). As the contradictory block is not obtained, the procedure evaluate is called on $\{\{3'\}\}$ and the solution $\{1',2,3'\}$ is found.

9. Tseitin's satisfiable formulae

9.1. DPL reduction of blocks

Property 9.1. *Let $\underline{c}$ be a subset of $\underline{A}$, $\underline{i}$ be an atom of $\underline{c}$ and ε be a sign. Under those conditions, $Ri(\mathscr{B}^{\varepsilon}(\underline{c})) = \mathscr{B}^{-\varepsilon}(\underline{c}\setminus\{\underline{i}\})$ and $Ri'(\mathscr{B}^{\varepsilon}(\underline{c})) = \mathscr{B}^{\varepsilon}(\underline{c}\setminus\{\underline{i}\})$.*

Proof. By the second replacement rule, $\mathscr{B}^{\varepsilon}(\underline{c})) = \{i\}\cdot\mathscr{B}^{\varepsilon}(\underline{c}\setminus\{\underline{i}\})\cup\{i'\}\cdot\mathscr{B}^{-\varepsilon}(\underline{c}\setminus\{\underline{i}\})$. The result follows from the definition of DPL reduction.

Property 9.2. *Let G be an alternated graph and $\underline{i}$ be the index of an edge. Under those conditions, $Ri(C(G)) = C(G\setminus i')$ and $Ri'(C(G)) = C(G\setminus i)$.*

Proof. Let x be an end of $\underline{i}$. If $\underline{i}$ is not a loop then $G(x) = \mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x))\cdot(\mathscr{B}^{+}(\mathrm{fi}(x))\cup\mathscr{B}^{-}(\mathrm{fi}(x))) = \{i\}\cdot\mathscr{B}^{-\sigma(x)}(\mathrm{fe}(x)\setminus\{\underline{i}\})\cdot F\cup\{i'\}\cdot\mathscr{B}^{\sigma(x)}(\mathrm{fe}(x)\setminus\{i\})\cdot F$ [with $F = \mathscr{B}^{+}(\mathrm{fi}(x))\cup\mathscr{B}^{-}(\mathrm{fi}(x))$] $= \{i\}\cdot(G\setminus i)(x)\cup\{i'\}\cdot(G\setminus i')(x)$.

If $\underline{i}$ is a loop this equality is trivial. So if x is an end of $\underline{i}$, $G(x) = \{i\}\cdot(G\setminus i)(x)\cup\{i'\}\cdot(G\setminus i')(x)$ and the result follows from the definition of the DPL reduction.

Theorem 9.3 (Even graph theorem). *Let G be a graph. If all the connected components of G are even, the meta-resolution algorithm gives a solution of $C(G)$ in time $O(|C(G)|^2\cdot|U|^2)$.*

Proof. In that case $C(G)$ is satisfiable and the procedure evaluate is called recursively. We first prove by induction that the number of recursive calls is less than $2\cdot|U|$.

Basis. If $|U| = 1$ then either $R1(C(G))$ is empty and there is only one call or $R1(C(G))$ is equal to the contradictory block and there is a second call on $R1'(C(G))$.

Induction step. At the first (non-recursive) call, $C(G)$ being satisfiable the contradictory block is not obtained and a first recursive call holds on $R1(C(G)) = C(G\backslash 1')$ which may be satisfiable or not. If it is satisfiable then we conclude by the induction hypothesis. Otherwise by the odd graph theorem, the contradictory block is obtained

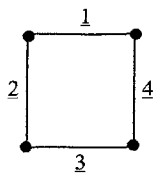


Fig. 14. A graph G.

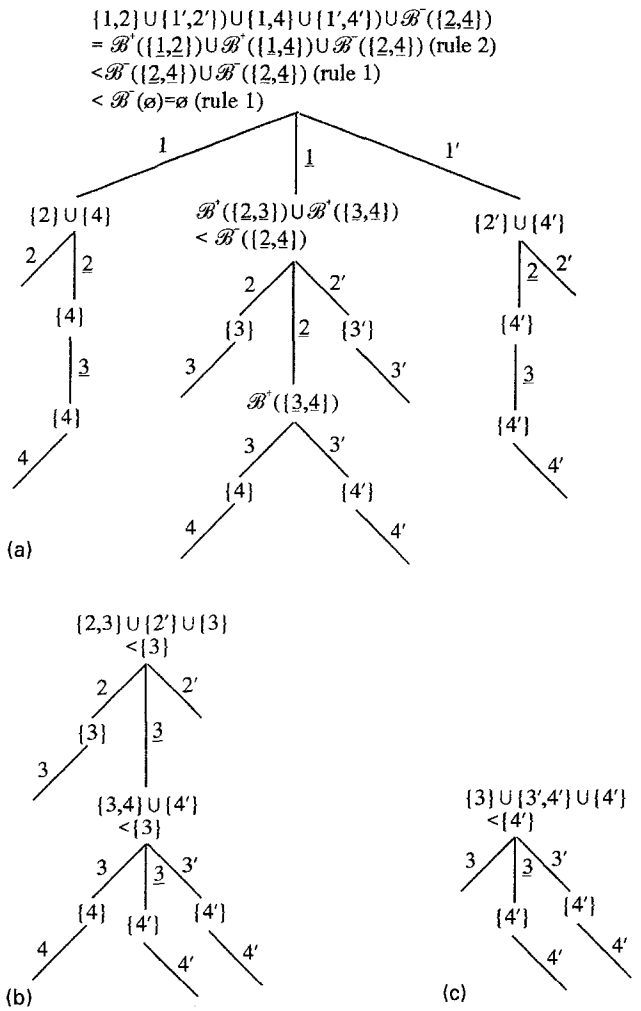


Fig. 15.

by meta-resolution and this first recursive call is a leaf of recursion. In that case, a second recursive call holds on $R1'(C(G)) = C(G\backslash 1)$ which is satisfiable and we conclude by the induction hypothesis.

Let n denote $|U|$ and m denote $|C(G)|$. Each recursive call computes a meta-resolution tree in time $O(m^2 \cdot n)$ by the odd graph theorem and at most two DPL reductions in time $O(m)$. The number of calls being less than $2n$, the solution is given in time $O(m^2 \cdot n^2)$.

Example. Let G be the even graph given by Fig. 14. The algorithm first constructs the tree of Tseitin's satisfiable formula $C(G)$ which is equal to $\{\{1,2\},\{1',2'\},\{1,4\},\{1',4'\},\{2,3\},\{2',3'\},\{3,4\},\{3',4'\}\}$ (see Fig. 15(a)).

As the contradictory block is not obtained, the procedure evaluate is called on $R1(C(G)) = C(G\backslash 1') = \{\{2'\},\{4'\},\{2,3\},\{2',3'\},\{3,4\},\{3',4'\}\}$ and constructs the tree given in Fig. 15(b).

As the contradictory block is not obtained, the procedure evaluate is called on $C(G\backslash\{1',2'\}) = \{\Lambda,\{4'\},\{3\},\{3,4\},\{3',4'\}\}$ whose contradiction is trivial.

So, the procedure evaluate is called on $C(G\backslash\{1',2\}) = \{\{4'\},\{3\},\{3,4\},\{3',4'\}\}$ and constructs the tree of Fig. 15(c).

As the contradictory block is not obtained, the procedure evaluate is called on $C(G\backslash\{1',2,3'\}) = \{\{4'\}\}$ and the solution $\{1,2',3,4'\}$ is found.

10. Conclusion

Since its introduction by Robinson [12], resolution has been the starting point of many algorithms. Most of them, designed for practical reasons, add to basis resolution a strategy for the choice of clauses (see [10] or [2]).

On the other hand, many efforts have been made to prove its exponentiality. A curious result of these works is the following: A proof of exponentiality on a given family of examples requires meta-arguments which can be integrated in new efficient algorithms. This fact was first formalized by Tseitin [13] when he introduced extended resolution.

In our own proof [6, 7], we use an implicit description of a resolution tree. This implicit description when applied to a Davis–Putnam tree, becomes a meta-resolution tree. Both extended resolution, which allows the introduction of new literals, and meta-resolution, which allows the introduction of new rules, enhance resolution by the use of meta-arguments.

Meta-resolution is a deterministic method to solve Tseitin formulae more efficiently than extended resolution. However, the important point is that we give a new illustration of the fact that constructive proofs of exponentiality cannot be used in front of the coNP–NP question.

References

[1] C. Berge, *Théorie des graphes et ses applications* (Dunod, Paris, 1958).
[2] M. Bezem, Completeness of resolution revisited, *Theoret. Comput. Sci.* **74** (1990) 227–237.
[3] V. Chvatal and E. Szeremedi, Many hard examples for resolution, J. ACM **35** (4) (1988) 759–768.
[4] S.A. Cook and R. Reckhow, The relative efficiency of propositional proofs systems, *J. Symbolic Logic* **44** (1979) 36–50.
[5] M. Davis and H. Putnam, A computing procedure for quantification theory, *J. ACM* **1** (1960) 201–215.
[6] J.D. Fouks, Sur la résolution du problème de satisfiabilité, Thèse n° 175, Université de Mulhouse, 1991.
[7] J.D. Fouks, Tseitin's Formulas revisited, *Theoret. Comput. Sci.* **99** (1992) 315–326.
[8] Z. Galil, On the complexity of regular resolution and the Davis–Putnam procedure, *Theoret. Comput. Sci.* **4** (1977) 23–46.
[9] A. Haken, The intractibility of resolution, *Theoret. Comput. Sci.* **39** (1985) 297–308.
[10] D.W. Loveland, *Automatic theorem prover, A logical basis* (North-Holland, Amsterdam, 1978).
[11] G. Margulis, Explicit construction of concentrators, *Problems Inform. Transmission* **9** (1973) 325–332.
[12] J.A. Robinson, A machine oriented logic based on the resolution principle, *J. ACM* **12** (1965) 23–41.
[13] G.S. Tseitin, On the complexity of derivations in the propositional calculus, in: A.O. Slisenko, ed., Structures in Constructive Mathematics and Mathematical Logic, Part II (Consultants Bureau, New York 1970) p. 115–125.
[14] A. Urquhart, Hard examples for resolution, *J. ACM* **34** (1) (1987) 209–219.